Sistema de Gestión de Talento Humano

Las Brisas

Autor: Carol Nicole Marentes Torres

SENA

Tecnólogo en Análisis y Desarrollo de Software  
Correo: carolmarentes31@gmail.com  
Fecha: 2025

Neiva - Huila

# Resumen

Se registra el diseño, la implementación y la evaluación inicial de una plataforma web de Gestión de Talento Humano para la empresa “Las Brisas”. La propuesta combina autenticación con control de roles, módulos de empleados y contratos, registro de asistencia, gestión de permisos e incapacidades, capacitaciones/inducción, y generación de certificados laborales. La arquitectura se basa en el API REST con Spring Boot y un frontend en React, y persistencia en MySQL. Se siguió un enfoque iterativo con pruebas unitarias e integración; se incluyen diagramas UML, MER/ER. La propuesta se justifica en bibliografía sobre frameworks frontend, desarrollo Java empresarial, diseño responsivo y digitalización de RR. HH. [1][10][5][9]. Los resultados muestran mejora en tiempos de respuesta y trazabilidad, así como una base para analítica y futura migración a microservicios [12].

# Palabras claves

Gestión de Talento Humano.

Digitalización.

Spring Boot.

React.

JWT.

MySQL.

Analítica.

# 

# Introducción

El marco competitivo del sector cárnico impone procesos de talento humano ágiles, trazables y accesibles a través del tiempo. En “Las Brisas”, la se contaba con información tanto en archivos como en planillas teniendo en cuenta que había duplicidades y muy poco control del cambio de información. La digitalización de estos procesos hace posible eliminar reprocesos, y acelerando así la emisión de certificados y mejorar el cumplimiento normativo. La literatura revisada apunta a beneficios por la adopción de soluciones digitales en RR. HH.,[8] además de los inconvenientes por el proceso de adopción o, en su defecto, cambio cultural [9].

En cuanto al nivel tecnológico, se eligió un stack suficientemente soportado: React en el frontend por su arquitectura basada en componentes y su ecosistema, y Spring Boot en el backend por su madurez, seguridad y productividad en entornos empresariales [1][10]. En su interfase, se optó diseño responsivo de acuerdo con recomendaciones para la accesibilidad y lógica en el uso de dispositivos [5]. La opción de MySQL se validó a partir principios de modelado relacional y normalización [6]. Finalmente, se espera que se pueda ir evolucionando hacia una ruta de microservicios para módulos con alta lectura [12].

# Marco teórico

1. La Gestión de Talento Humano (GTH) en el entorno digital.

La GTH comprende los procesos de atracción, incorporación, desarrollo, compensación y retención del talento. En consecuencia, las soluciones digitales impulsan el seguimiento de los indicadores y contribuyen a una reducción en tiempos de procesos repetitivos. Algunos estudios revelan cómo el reclutamiento de personal, la formación y las mejoras en el desempeño se incrementan teniendo como punto de partida la interconexión entre plataformas y datos [9]. En el caso de las pymes, el impacto es mayor debido a la no existencia de sistemas legados por defecto y la llegada de la posibilidad de implantar procesos estándar.

1. Las modernas arquitecturas web.

Las aplicaciones web de la empresa han dado un giro hacia lo que son las arquitecturas de servicios existentes, con capas claramente definidas (interfaz de presentación(lógica), lógica y datos). En esta línea, el Spring Framework y el ecosistema que lo envuelve (Spring Boot, Spring Security, Spring Data...) son los que dominan el panorama de productividad y mantenibilidad en el mundo de Java, al igual que el soporte nativo que proporciona la seguridad, la gestión de datos o las pruebas de software [10][11]. React, en contraposición, permitiendo estructurar la interfaz de usuario mediante componentes y un ciclo de vida declarativo, favorece la reusabilidad y testabilidad de la misma [1].

1. Diseño responsivo y experiencia de usuario.

Bootstrap proporciona un sistema de rejillas, utilidades y componentes que simplifican y agilizan la construcción de interfaces con un grado de consistencia en distintas dimensiones de pantalla; al mismo tiempo que huimos de la aparición de deuda técnica e incrementamos la accesibilidad [5]. La evidencia sugiere que los sistemas adoptados con la interfaz de usuario con una misma estructura, con la existencia de un mismo feedback claro, traen consigo el incremento del uso de la misma.

1. Modelado de datos relacional.

Las claves primarias y foráneas, la normalización y el control de integridad permiten garantizar la calidad de los datos y evitar las anomalías de actualización. La literatura indica que un buen diseño conceptual y lógico reduce los costes de mantenimiento y favorece la analítica posterior [6].

1. Visualización y analítica operacional.

Las bibliotecas Google Charts permiten construir paneles de control que miden la asistencia, los permisos y la rotación de personas, además la integración de estos con fuentes relacionales es un patrón común en los mismos tableros operativos [6].

1. Microservicios y estandarización.

La estandarización de contratos de servicio, la observabilidad y el versionado son esenciales durante la migración a microservicios. Informes de casos de estudio evidencian mejoras en latencias para servicios de lectura de gran carga mediante la especialización de los mismos [12]. En pymes, una traslación progresiva es más recomendable puesto que limita los riesgos y los costes.

## Competencias técnicas y ecosistema de lenguajes.

## La elección del lenguaje de programación viene respaldada por el mercado (Java y JavaScript son soportados por la demanda, se reproduce la disponibilidad de proyectos y librerías y también la comunidad, podría justificar su uso en proyectos académicos o empresariales [8]). React Native y Flutter son alternativas para construcciones móviles, sin embargo, para intranets y flujos administrativos, una SPA web permitirá tener la menor fricción de despliegue [3].

## 2.1 React y componentes

## React establece componentes puros, elevación de estado y composición; estas prácticas favorecen pruebas unitarias, mocking y separación de responsabilidades que repercute en mantenibilidad [1].

## 2.2 Spring Boot y ecosistema

Spring Data simplifica repositorios, mientras que Spring Security habilita políticas declarativas. La configuración 'convention over configuration' acorta el tiempo hasta producción [10].

## 4.1 Normalización en el dominio RR. HH.

## Las dependencias parciales/transitivas y las claves compuestas se tratan aplicando 2FN/3FN, documentando las reglas de negocio para la integridad referencial y los catálogos maestros (áreas, cargos) [6].

## 5.1 Analítica operativa incrementable

Al iniciar con métricas descriptivas (presentismo, permisos) y evolucionar a analítica diagnóstica/predictiva, se minimiza el costo inicial y se maximizan resultados tempranos [6].

## 6.1 Estrategias de transición a microservicios

Se recomienda extraer primero servicios con alta lectura/aislamiento claro (asistencia, notificaciones), con gateways y observabilidad básica [12].

# Métodos

Diseño metodológico. Se utilizó un proceso iterativo-incremental con hitos funcionales. Cada sprint definió historias de usuario, criterios de aceptación y casos de prueba. El diseño se apoyó en UML (casos de uso, secuencia) y un MER/ER que estableció entidades, relaciones y restricciones [6].

Arquitectura y seguridad. El backend implemento API REST usando Spring Boot controller–service–repository, DTOs, Spring Security con JWT. Las políticas CORS, el cifrado de contraseñas y la auditoría de cambios preservan confidencialidad e integridad [10].

Frontend. React organiza la UI en componentes reutilizables y está formado por formularios controlados, validaciones en cliente y navegación por rol. [1][5].

Persistencia de datos. Se creó el esquema relacional en MySQL para usuarios, roles, empleados, contratos, horarios, asistencias, solicitudes (permisos/incapacidades), capacitaciones/inducciones, y se definió la estrategia de índices y claves foráneas según patrones de consultas [6].

Pruebas y documentación. Se documentaron endpoints mediante Swagger. Como referencia académica, se consultaron estudios de aulas/laboratorios con React/Spring Boot, y experiencias de integración de gráficos con bases relacionales [14][6].

# Resultados

1. 1) Autenticación y perfiles. Inicio de sesión con JWT; renovaciones de token; recuperación de contraseña; perfil del empleado con datos personales, área/cargo/ubicación y contrato activo. [10][6][9]
2. 2) Asistencia. Registro de entrada/salida, cálculo de horas trabajadas, gestión de turnos. [10][6][9]
3. 3) Permisos e incapacidades. Solicitud con tipo, fechas, motivo y adjuntos; flujo de estados; observaciones. [10][6][9]
4. 4) Capacitaciones/inducción. Asignación de contenidos, evaluaciones, seguimiento de avances y registro de participación. [10][6][9]
5. 5) Contratos. Catálogo de tipos, fechas de inicio/fin, renovaciones y alertas por proximidad a vencimiento. [10][6][9]
6. 6) Certificados. Generación de certificados laborales con datos parametrizables. [10][6][9]
7. 7) Reportes y visualizaciones. Indicadores operativos y tendencias de asistencia/permisos. [10][6][9]
8. 8) Auditoría y seguridad. Registro de acciones (crear/actualizar/eliminar), control de roles y verificación de integridad. [10][6][9]

Evaluación preliminar. Con usuarios de prueba, se observó reducción en tiempo de emisión de certificados y consolidación de asistencia, además de menores inconsistencias por duplicidad de archivos. [9]

|  |  |  |
| --- | --- | --- |
| Módulo | Evidencias de funcionalidad | Referencias |
| Autenticación/JWT | Login, autorización por rol, expiración/refresh | [10] |
| Empleados/Contratos | Ficha, contratos, fechas claves | [10] |
| Asistencia | Entradas/salidas, turnos, reportes | [14] |
| Permisos/Incapacidades | Flujo estados, adjuntos, bitácora | [15] |
| Capacitaciones/Inducción | Asignación, avances | [9] |
| Reportes/Analítica | Gráficos, tendencias | [6] |

# Discusión

Elección del stack. El tándem Spring Boot + React maximizó productividad y separación de responsabilidades; la evidencia comparativa apoya esta decisión en escenarios empresariales y educativos [10][11].

Experiencia de usuario. Un diseño responsivo consistente reduce curva de aprendizaje y errores, en línea con lo reportado para Bootstrap y patrones de UI modernos [5][1].

Estrategia de datos. Un modelo normalizado favorece integridad y consultas reproducibles para analítica [6].

Escalabilidad. La estandarización para microservicios sugiere ganancias considerables en escenarios de lectura intensiva; no obstante, se recomienda un paso gradual para evitar complejidad prematura [12].

Limitaciones. Datos maestros en consolidación (áreas, cargos), reportes analíticos iniciales, pruebas de carga pendientes y definición de políticas de retención documental.

# Conclusiones

El sitio web de “Las Brisas” agrupa los procesos relacionados con el Talento Humano, haciendo mejoras en la trazabilidad y disminuyendo tiempos operativos. El diseño modular y el uso de estándares (REST, JWT) sientan bases para analítica [12]. Próximos pasos: paneles ejecutivos de RR. HH., módulo de vacaciones, firma electrónica y ampliación de notificaciones.

# Conflictos de interés / Declaración ética

No existen conflictos de interés. Se usaron datos de prueba y se cuidó la protección de información personal.

# Agradecimientos

A instructores del SENA y compañeros por la revisión técnica y apoyo durante pruebas funcionales.

A mi familia y red de apoyo por el acompañamiento emocional y logístico durante todo el ciclo del proyecto, cuyo respaldo hizo posible sostener los periodos de estudio, implementación y documentación necesarios. Finalmente, a todas las personas que, de forma directa o indirecta, contribuyeron con referencias, mejora de estilos y pruebas; este logro refleja un esfuerzo colectivo.

# Referencias

* [1] Análisis de Frameworks Frontend para Aplicar UX/IU en el Desarrollo Web: Una Revisión Sistemática. (2025, August 21). Recuperado de https://dialnet.unirioja.es/servlet/articulo?codigo=9589652 — Google Scholar.
* [2] Práctica en lenguaje de programación Java y nuevas tecnologías. (2025, August 21). Recuperado de https://repositorio.utp.edu.co/entities/publication/6fd03704-13f6-4da5-a8cf-f43bd92dc34d — Google Scholar.
* [3] React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma. (2025, August 21). Recuperado de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.revista.unam.mx/wp-content/uploads/v20\_n5\_a5\_React-Native-acortando-las-distancias-entre-desarrollo-y-dise%C3%B1o-m%C3%B3vil-multiplataforma.pdf — Google Scholar.
* [4] ESTUDIO COMPARATIVO DE LOS FRAMEWORKS DEL DESARROLLO MÓVIL NATIVO "FLUTTER" Y "REACT NATIVE".. (2025, August 21). Recuperado de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://dspace.utb.edu.ec/bitstream/handle/49000/10516/E-UTB-FAFI-SIST-000242.pdf?sequence=1&isAllowed=y — Google Scholar.
* [5] Optimización web móvil: El poder de bootstrap en el desarrollo adaptativo. (2025, August 21). Recuperado de https://revistas.milpaalta.tecnm.mx/index.php/IPSUMTEC/article/view/337/619 — Google Scholar.
* [6] Análisis de la conexión de Google charts a bases de datos y otras fuentes para visualización dinámica de datos. (2025, August 21). Recuperado de https://revista.gnerando.org/revista/index.php/RCMG/article/view/690/724 — Google Scholar.
* [7] Bases de datos. (2025, August 21). Recuperado de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://repositorio.uchile.cl/bitstream/handle/2250/151632/Bases-de-datos.pdf?sequence=1&isAllowed=y — Google Scholar.
* [8] Análisis de los lenguajes de programación más utilizados en el desarrollo de  
  aplicaciones web y móviles. (2025, August 21). Recuperado de https://dialnet.unirioja.es/servlet/articulo?codigo=8635309 — Google Scholar.
* [9] GESTIÓN DEL TALENTO EN LA ERA DIGITAL: CÓMO ATRAER, RETENERY POTENCIAR PROFESIONALES EN LA EDUCACIÓN DEL SIGLO XXI. (2025, August 21). Recuperado de https://revistas.qlu.ac.pa/index.php/latitude/article/view/228/168 — Google Scholar.
* [10] ESTUDIO DE LAS APLICACIONES WEB EMPRESARIALES, DESARROLLADAS EN EL LENGUAJE DE PROGRAMACIÓN JAVA, EN LOS FRAMEWORKS HIBERNATE Y SPRING. (2025, August 21). Recuperado de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://dspace.utb.edu.ec/bitstream/handle/49000/9528/E-UTB-FAFI-SIST-000211.pdf?sequence=1&isAllowed=y — Google Scholar.
* [11] ANÁLISIS COMPARATIVO DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES WEB EN JAVA. (2025, August 21). Recuperado de https://revistas.uss.edu.pe/index.php/ING/article/view/101/100 — Google Scholar.
* [12] Implementación y estandarización en microservicios para altos consumos de lectura en sistemas transaccionales. (2025, August 21). Recuperado de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://bibliotecadigital.udea.edu.co/server/api/core/bitstreams/abaeb5aa-1785-4304-85e2-98229d666c78/content — Google Scholar.
* [13] SOLUCIONES DIGITALES PARA LA GESTIÓN DE RECURSOS HUMANOS, EN LOS ÚLTIMOS DIEZ AÑOS. (2025, August 21). Recuperado de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://repositorio.upn.edu.pe/bitstream/handle/11537/25670/Casas%20Perea%2C%20Jose%20Carlos.pdf?sequence=1&isAllowed=y — Google Scholar.
* [14] Laboratorio Basado en Web de Estadísticas y Probabilidad Multimedia con Spring Boot y React.js de apoyo a la enseñanza. (2025, August 21). Recuperado de https://terc.mx/index.php/terc/article/view/263/239 — Google Scholar.
* [15] Análisis, diseño e implementación de aplicación Front-end Manual de Apoyo de Procesos de la Facultad de Ingeniería (MAPI). (2025, August 21). Recuperado de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://bibliotecadigital.udea.edu.co/server/api/core/bitstreams/2ff59c7b-57ff-4dad-abb8-ed4347fbe192/content — Google Scholar.